and Fig. 6 [7]B is a graph illustrating the number of local interactions in the system according to the present invention as a function of the frame number, which corresponds to the elapsed time;

[Fig. 8 is a diagram illustrating an example of an algorithm of a version optimized in terms of a block diagonal system;]

Fig. 7 [9] is a table illustrating the evaluation results in terms of the stability and the calculation load during a game, for the algorithm according to the present invention;

Fig. 8 [10] is a schematic diagram illustrating one scene of an animation of a group of bees produced according to the present invention;

Fig. 9 [11] illustrates examples of matrices used to generate the animation shown in Fig. 8 [10];

Fig. 10A [12A] is a graph illustrating the evaluation function (optimal cost) for another example of an animation as a function of the frame number, and Fig. 10B [12B] is a timing chart illustrating the changes in the state space in terms of the roles assigned to six airplanes contained in an formation of airplanes used in the example shown in Fig. 10A [12A];

Fig. 11A [13A] is a graph illustrating the performance index associated with another example of an animation according to the present invention as a function of the frame number, and Fig. 11B [13B] is a timing chart illustrating the changes in the state space in terms of the roles assigned to six fishes contained in a group used in the example shown in Fig. 11A [13A];

[Fig. 14 is a schematic diagram illustrating an example of an off-line algorithm for calculating parameters associated with a coordination process according to the

present invention;]

[Fig. 15 is a schematic diagram illustrating another example of an off-line algorithm for calculating parameters associated with a coordination process according to the present invention;]

Fig. 12 [16] is a functional block diagram of an entertainment apparatus using a control method according to the present invention;

Fig. 13 [17] is a schematic diagram illustrating an example of the hardware configuration of an entertainment apparatus using the control method according to the present invention; and

Fig. 14 [18] is a circuit diagram illustrating an example of a circuit configuration of the entertainment apparatus using the control method according to the present invention.

Please make the following amendments to page 34, lines 1-19:

An example of a novel algorithm for achieving coordination at run time according to the present invention is shown below [in Fig. 6]. This example of the algorithm [shown in Fig. 6] has a high responsiveness and may be performed in real time. Furthermore, at the design stage, the optimum feedback rule is determined from the characteristics of the dynamic system according to the evaluation criteria using an off-line algorithm including a procedure widely used to solve the LQ problem. This will be discussed in further detail later in Section (A-8).

RUN-TIME COORDINATION ALGORITHM (A1)

Exert Continuous Multivariable Controls

(Role Assignments Defined by π_n)

1. $U_n \leftarrow -G[X_n - \pi_n . X_{ref}]$

Dynamic Simulation Step

2. $X_{n+1} \leftarrow f(X_n, U_n)$

3. $J \leftarrow (X_n - \pi_n . X_{ref})^T S (X_n - \pi_n . X_{ref})$

Sort Entities According to their Contribution to Overall Cost

4. For each entity j do

5. $X_n^{-j} \leftarrow [X_n^1, X_n^2, \ldots, j^{th}$ coordinate of $(\pi_n X_{ref}), \ldots, X_n^N]$

6. $J_j(n) \leftarrow (X_n^{-j} - \pi_n . X_{ref})^T S (X_n^{-j} - \pi_n . X_{ref})$

7. Endfor

8. Sort entities by decreasing $J_j(n)$

Look for Opportunistic Role Swaps in K Worst Performers

9. For K! permutations $\pi(k)$ of K candidates do

10. $J(\pi(k)) \leftarrow (X_n - \pi(k) . X_{ref})^T S (X_n - \pi(k) . X_{ref})$

11. if $(J^* > |J(\pi(k)) + \Delta_{min}|)$ then

12. $J^* \leftarrow J(\pi(k))$

13. $k^* \leftarrow k$

14. endif

15. endfor

Update Role Assignments Switchboard

16. $\pi_{n+1} \leftarrow \pi(k)^*$

---

1. Feed-forward gains (or integral terms) ensure disturbance rejection.

2. Threshold prevents flickering due to non-linear dynamics.

3. Permutations are computed beforehand (see offline algorithm in Section (A-8)).

4. Role assignment is represented by matrix $f\hat{\imath}$ for convenience, yet implementation relies on a simple indirection rather than matrix multiplication.

The algorithm shown above [in Fig. 6] is applied for each time step. The calculation efficiency of the algorithm

may be increased by various methods including a clustering technique used in a simulation of a many-body (n-body) problem or a method adapted to characteristics intrinsic to a particular case. Although a matrix representation is employed, the matrix representation is not always efficient.

Please make the following amendments to page 38, line 6 through page 39, line 22:

Although Liapunov's method can be applied to nonlinear dynamics, the execution thereof is complicated and it is necessary to perform a computer simulation to check whether the algorithm is trapped. Fig. 6A [7A] is a graph showing the performance index, and Fig. 6B [7B] is a graph showing the number of local interactions. In Fig. 6B [7B] showing the number of local interactions, open bars represent the number of collisions and solid bars represent the number of role replacements. In both Figs. 6A [7A] and 6B [7B], the horizontal axis represents the frame number which corresponds to the elapsed time. It can be seen from Figs. 6A [7A] and 6B [7B]that the number of collisions and the performance index both decrease with increasing frame number and with increasing number of role replacements.

(A-5-3) Real-Time Performance

The algorithm (A1, above; and A2, below [Figs. 6 and 8]) according to the present invention has been evaluated in terms of the stability and the calculation load imposed upon a game under the conditions of a frame rate of 50 Hz and a peak number of instructions of the order of GFLOPS/sec (Fig. 7 [9]). The algorithm has been executed on the PlayStation 2 produced by Sony Computer Entertainment, Inc. It has been assumed that the CPU time spent for the processing associated with the dynamic characteristics and the coordination is very

short (of the order 1% of the total CPU time), and the number of members whose roles were replaced was limited to K = 6. When the number N of members is small, the processing associated with the coordination and the processing associated with the dynamic characteristic need similar calculation loads. However, the processing associated with the coordination becomes dominant when N is large.

The algorithm A2 shown below [in Fig. 8] is an example of an algorithm of a version optimized in terms of the block diagonal system. The algorithm A2 shown below [in Fig. 8] has better linear characteristics than the algorithm A1 shown in Section (A-5) [Fig. 6], because the core matrices S_∞, A, B, and G in the algorithm A2 increase in size linearly as N increases, while they increase in size quadratically in the algorithm A1.

OPTIMIZED VERSION FOR BLOCK-DIAGONAL SYSTEMS (A2)

1. $U_n \leftarrow -G[X_n - \pi_n . X_{ref}]$
2. $X_{n+1} \leftarrow f(X_n, U_n)$
3. For j=1 to N do
4. $\delta J(i,j) \leftarrow [X_n^j - (\pi_n . X_{ref})^j]^T S^{entity} [X_n^j - (\pi_n . X_{ref})^j]$
5. endfor
6. Sort entities by decreasing $\delta J(i,I)$ & select first K ones
7. for k=1 to K do
8. for p=k to K do
9. $\delta J(k,p) \leftarrow [X_n^K - X_{ref}^p]^T S^{entity} [X_n^K - X_{ref}^p]$
10. endfor
11. endfor

Look for Opportunistic Role Swaps in K worst performers

12. For K! permutations $\pi(k)$ of K candidates do
13. $J(\pi(k)) \leftarrow \sum_{p=0}^{N} \delta J(p, -X_{ref}(p,\pi(k)) \text{ or target}(p, \pi(k)))$

14. if ($J^* > |J(\pi(k))+\Delta_{min}|$) then

15. $J^* \leftarrow J(\pi(k))$

16. $k^* \leftarrow k$

17. endif

18. endfor

19. $\pi_{n+1} \leftarrow \pi(k^*)$

(1) Sorting candidates for role swapping on a simple distance-to-target criterion corresponds to replacing Sentity by the identity matrix.

Please make the following amendments to page 41, line 5 through page 42, line 8:

By way of example, an animation of a group including 50 members (bees) is described. In this example, the bees in the group behave as a group (have linear attraction), while they behave repulsively when they come very close to each other (that is, there is nonlinearity and the Newton's law of collision holds). One scene of an animation of the group of bees is shown in Fig. 8 [10].

In Fig. 8 [10], 50 bees gather within a particular region below light. Although a dynamic model employed herein is not a mechanical model employed in many other cases, the dynamic model employed allows suppression of acceleration of expansion of members. Thus, a certain systematic group behavior is described by this dynamic model.

The system equation is given as $X_{n+1} = AX_n + BU_n + (X_n)$. Matrices such as those illustrated conceptually in Fig. 9 [11] are required for the calculation according to the algorithm A1. In Fig. 9 [11], the matrix G is used to provide inputs of members and the matrix S is used to determine the cost (performance index) such as a simple

distance. In Fig. 9 [11] , non-zero elements in the matrices are represented by shaded areas, and zero elements are represented by open areas. For a group which includes a large number of members, as is the case with the present group of bees, the calculation load associated with the multivariable feedback becomes greater than that of the simulation. This can also be seen from comparison between the sizes of matrices A and B and the sizes of matrices G and S_∞.

Please enter the following amendments to page 42, lines 16-24:

As shown in Fig. 6 [7], although the group is collapsed at an initial stage, bees gradually gather into a specified form after a layout is applied when the frame number = 21. In the present example, the number of role replacements is limited to 6 for each frame. Because the replacement of roles needs a finite period of time, a greater number of members of the group can be involved in practice. In spite of collisions, the group of bees forms a specified shape.

Please enter the following amendments to page 43, line 15 through page 44, line 6:

Fig. 10A [12A] is a graph illustrating the evaluation function (optimal cost) for the present example as a function of the frame number. Fig. 10B [12B] is a timing chart illustrating the changes in the state space in terms of the roles assigned to six airplanes contained in the airplane formation of the present example. In the present example, the control rule is calculated upon the assumption that airplanes fly in formation and an operating point is obtained near the point at which the Jacobian of the system is extracted. The layout is changed at frame number = 100 and

also at frame number = 200 thereby triggering the replacement of roles among airplanes.

As can be seen from Fig. 10 [12], when the nonlinearity is not very strong, as is the case in the present example, the performance index generally reduces. The threshold value is reduced to about 30% in order to avoid flicker and sporadic replacement of roles.

Please enter the following amendments to page 44, line 26 through page 45, line 17:

The nonlinearities cause the state-space trajectories of fishes of the group to become complicated, as shown in Fig. 11 [13], and the algorithm functions in a manner which is very different from the predicted manner.

Fig. 11A [13A] illustrates the performance index in the present example as a function of the frame number. Fig. 11B [13B] is a timing chart illustrating the changes in the state space in terms of the roles assigned to six fishes contained in the group in the present example, wherein lines represent respective fishes and the vertical coordinate represents targets. When there is strong nonlinearity as is the case in the present example, the performance index decreases in a random fashion. For example, replacement of roles is triggered by a change in the layout at frame number = 300. Spontaneous replacement of roles occurs near frame number = 200.

Please enter the following amendments to page 50, lines 6-19:

At the design stage, the coordination method is described by parameters using a matrix obtained by two off-line algorithms. A first algorithm ("Multivariable LQ Control Synthesis", below [Fig. 14]) calculates the

multivariable optimum feedback on the basis of eigenvector expansion and solves the ARE (standard algorithm in LQ synthesis)

MULTIVARIABLE LQ CONTROL SYSTHESIS

1. Given the dynamic system A, B and performance index matrices Q, R, form the Hamiltonian matrix defined as:

$$H = \begin{bmatrix} B+AR^{-1}B^{T}A^{-T}Q & -BR^{-1}B^{T}A^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix}$$

2. Complete eigenvalues λ and eigenvectors W of H, collecting all couples such that the eigenvalue's modulus is smaller than one (stable pole of the closed-loop system)

3. Compute $S = \lambda W^{-1}$

4. Compute the optimal matrix K Gain as

$$G = [R+B^{T}S B]^{-1}B^{T}S A$$

The second off-line algorithm ("Function PermutationArray=Permute(vector)", below [Fig. 15]) recursively generates K! permutations and stores them in advance. These calculation results are sequentially used at run time to evaluate the role replacement. When K = 7, the number of entries of the table becomes 4,320. The permutation algorithm is described in R. Grimaldi, "Discrete an Combinational Mathematics", Addison-Wesley, New York, 1999.

Function PermutationArray=Permute(vector)

1. n=length(vector)

```
2.   if(n==1) then
3.      PermutationArray=vector
4.   else
5.      PermutationArray=NULL array
6.      Height=0
7.      for k=1 to n do
8.         sub-vector(k)=ShrinkVector(vector, k)
9.         SubArray=Permute(sub-vector)
10.        for i=1 to (n-1)! do
11.           for j=1 to n-1 do
12.              PermutationArray [i+Height][j]=SubArray[i][j]
13.           endfor
14.           PermutationArray [i+Height][n]=vector [k]
15.        endfor
16.        Height=Height+(n+1)!
17.     endfor
18.  endif
19.  return PermutationArray
```

Please enter the following amendments to page 51, lines 1-7:

Fig. 12 [16] illustrates, in functional block diagram form, main parts of the entertainment apparatus of the present embodiment.

In Fig. 12 [16] , reference numeral 101 denotes control means used by a game player to input various commands. Command data input via the control means 101 is transmitted to game executing means 102.

Please enter the following amendments from page 53, line 1 through page 54, line 23:

An example of the appearance and an example of the general hardware configuration of the entertainment apparatus

of the present embodiment are described below with reference to Figs. 13 [17] and 14 [18].

Fig. 13 [17] illustrates an example of the appearance of the entertainment apparatus. The entertainment apparatus 1 reads a game program stored on an optical disk such as a CD-ROM and executes it in response to an operation performed by a game player.

The main unit 2 of the entertainment apparatus 1 includes a disk loading part 3 in which an optical disk on which a game program is stored is loaded, a reset switch 4 for resetting the game, a power switch 5, a disk control switch 6 for controlling the loading of the optical disk, and slots 7A and 7B (two slots in this specific example).

The slots 7A and 7B are used to connect a controller 20. The slots 7A and 7B may also be used to connect a memory card so that game data is stored to or read from it.

The entertainment apparatus 1 is connected to a monitor 30 and loudspeakers 40, although connecting parts are not shown in Fig. 13 [17]. In the example shown in Fig. 13 [17], a television set is used as the monitor 30 and also as the loudspeakers 40.

Fig. 14 [18] illustrates the general hardware configuration of the entertainment apparatus 1. As shown in Fig. 14 [18], the entertainment apparatus 1 includes a control system 50 including a CPU 51 and associated peripheral devices, a graphic system 60 including a GPU 62 which writes an image into a frame buffer 63, a sound system 70 including an SPU (sound processing unit) 71 for producing musical sounds or sound effects, an optical disk control system 80 for controlling the optical disk on which the game program is stored, and a communication control system 90 for controlling the inputting of a signal output by the controller 20 in response to a command given by a game player

and also for controlling the inputting/outputting of data to/from a memory card 10 in which settings of the game or the like are stored.

In Fig. 14 [18], the control system 50 includes the CPU 51, a peripheral device controller 52 for controlling interrupt handling and direct memory access transfer operations, a random access memory (RAM) serving as a main memory 53, and a read only memory (ROM) 54 in which is stored an operating system (OS) for controlling various parts such as the main memory 53, the graphic system 60, and the sound system 70.

Please enter the following amendments to page 55, lines 13-19:

In Fig. 14 [18], the graphic system 60 includes a GTE (geometry transfer engine) 61 for performing a process such as a coordinate conversion, a GPU 62 for writing an image in accordance with an image write command given by the CPU 51, a frame buffer 63 for storing the image written by the GPU 62, and an image decoder 64 for decoding image data coded in a compressed fashion.

Please enter the following amendments to page 56, line 23 through page 57, line 22:

In Fig. 14 [18], the sound system 70 includes the SPU 71 for producing sounds such as musical sounds or sound effects in accordance with a command given by the CPU 51, and also includes a sound buffer 72 used by the SPU 71 to store waveform data. The musical sounds or sound effects produced by the SPU 71 is output to the loudspeakers 40. The SPU 72 further has the capability of decoding audio data and also the capability of reproducing, directly or after modulating, waveform data.